

We claim:

1. Method for the direct call of a target function by means of a start function through a processor with a memory management unit (MMU) in a computer operated by an operating system, characterized in that the start function is a component of a first task with a first memory context and the target function is in another memory context and in that the first task performs a context switch from the first memory context into the other memory context, the target function is executed, and this context switch is reversed after the execution of the target function.

10 2. Method according to Claim 1, characterized in that in order to perform the context switch, the physical address of the memory context of the task containing the target function is written into the MMU control register by the first task.

15 3. Method according to Claim 1, characterized in that parts of the first task as well as parts of the second task containing the target function are copied into a new memory context and the context switch is performed into this new memory context.

20 4. Method according to Claim 3, wherein the computer features working memory (RAM) and mass storage, e.g., a hard disk, and wherein the operating system swaps out parts of the working memory to the mass storage as required, characterized in that the memory region of the copied memory context is locked at least during the runtime of the target function called by the start function in order to avoid swapping out this memory region from the working memory.

25 5. Method according to claim 1, characterized in that the start function avoids task switching by the operating system during the runtime of the target function by deactivating the interrupt handling by means of a processor control register.

30 6. Method according to claim 1, characterized in that the target function includes no program steps containing a call of the operating system.

7. Method according to claim 1, characterized in that during the call to the target function by the start function, a new call to the target function by a function outside of the target function is blocked.

5 8. Method according to claim 1, characterized in that the task containing the target function only executes accesses that are performed entirely within a processing cycle to data that could be used or changed by the target function.

10 9. Method according to claim 1, characterized in that the task containing the target function only executes accesses that are protected by flags to data that could be used or changed by the target function.

15 10. Software program to be loaded into the working memory of a computer operated by an operating system with a processor with a memory management unit (MMU), characterized in that it includes a task with a start function for performing a method according to claim 1.

20 11. Software program according to Claim 10, characterized in that it includes a second task with a target function for performing a method according to claim 1.

25 12. Machine-readable data carrier with a software program which implements a method for the direct call of a target function by means of a start function through a processor with a memory management unit (MMU) in a computer operated by an operating system, characterized in that the start function is a component of a first task with a first memory context and the target function is in another memory context and in that the first task performs a context switch from the first memory context into the other memory context and this context switch is reversed after the execution of the target function.

30 13. A method for the direct call of a target function by means of a start function in a computer system comprising a processor with a memory management unit (MMU), wherein the computer system includes an operating system, wherein the start function is a component of a

first task with a first memory context and the target function is in another memory context, the method comprising:

the first task performing a context switch from the first memory context into the other memory context;

5 executing the target function in the other memory context;

reversing said context switch to return to the first memory context after executing the target function.

14. The method according to Claim 13, wherein the CPU includes a MMU control
10 register;

wherein the first task performing the context switch includes the first task writing the physical address of the memory context of the task containing the target function into the MMU control register.

15. The method according to Claim 13, further comprising
copying parts of the memory context of the first task as well as parts of the memory context of the second task containing the target function into a new memory context;
wherein the first task performing the context switch comprises the first task performing the context switch into the new memory context.